



TITLE:

ca.jsによる教材作成 (数学ソフトウェアとその効果的教育利用に関する研究)

AUTHOR(S):

松川, 信彦

---

CITATION:

松川, 信彦. ca.jsによる教材作成 (数学ソフトウェアとその効果的教育利用に関する研究). 数理解析研究所講究録 2019, 2105: 181-189

ISSUE DATE:

2019-02

URL:

<http://hdl.handle.net/2433/251889>

RIGHT:

# ca.js による教材作成

大阪府立佐野工科高等学校 松川 信彦

Nobuhiko Matsukawa, Osaka Prefectural Sano Technology High School

## 1 はじめに

大学レベルの数学の特定のトピックを題材に、クライアントサイドの JavaScript のみで、その理解を深めるための実例生成型の web 教材の作成を目標とする。この稿では、特に Jordan 標準形を主題にした教材開発の実践を紹介する。web 教材といえば、moodle などの LMS でよく見かけられるような、ドリル問題の解答定着型のコンテンツが主流であるが、数学科の学生の勉強方法である「行間を丁寧に追う」「実例と対比して理解を深める」姿勢の定着までカバーできる教材を理想としている。一方で、web ブラウザ上で閲覧できる数学のテキスト類は、大抵 PDF、あるいは MathJax を利用したものいずれかであり、数学の文書を読む訓練を受けていない学生、あるいは社会人のために、何らかの CAS と連動し、わかり易くする仕掛けを施してあるものは数が少ない。実例を自動生成し、主要な定理などと照らし合わせながら理解を深めるコンテンツといえば、Mathematica のノートブックのように、商用の CAS を利用したダウンロードコンテンツもあるが、無償かつ、作成・公開・実施が手軽という点で web 教材は大変有益であると思われる。

近年、機械学習の流行と相まって、システムエンジニアやプログラマーを中心に社会人向けの大学レベルの数学の勉強会が、首都圏や近畿圏などで定期的に開催されている。そのような勉強会の一つである Math & Coding [8] では、確率統計や強化学習などの実務に直結する勉強会の他にも、集合・位相や測度論、線形代数などの数学科の初年級レベルの勉強会も開かれている。著者は数学に興味を持つエンジニアとの交流を目的に、線形代数の勉強会に幾度か参加したことがある。教科書は [11] が用いられた。工学部出身の参加者が多かったためか、商空間や Jordan 標準形などの抽象度の比較的高い内容は難易度が高いとのことで、勉強会では割愛するとのことであった。確かに、教科書 [11] においては、一般固有空間を用いた Jordan 標準形の構成方法は複雑に入り組んでおり、読み解くための労力は並大抵ではない。Jordan 標準形は、大学時代に線形代数学で受講したものの、結局よく理解せぬままであったという参加者もいた。大学初年級の学生を対象とした [11] において、単因子論が避けられたのは教育的配慮であろうと思われるが、単因子論を利用した方が、学習コストを大幅に低減でき、理論の見通しもすっきりするものと著者は考えている。特に、著者は JavaScript の CAS の ca.js [3] を開発しており、整数や多項式環上の行列に対し、単因子を計算するプログラムを実装していたところであり、実例の計算を以って、プログラマーである彼らの理解に寄与できるものと確信していた。「Jordan 標準形ショートカット」と題し、1時間の講義を行った。ホワイトボードでは理論を展開し、各々が持参している端末から私のサイトに誘導し、web ブラウザの JavaScript で計算した結果と照らし合わせて確認してもらう方法をとった。「わかり易

い」「実際に計算が動いている様子が生々しい」「濃い話だった」など、肯定的な反応が得られた。

## 2 ca.js について

ca.js とは、著者が JavaScript の多倍長整数ライブラリである BigInteger.js [1] のみを利用して、「素朴な JavaScript」だけで、標準的な計算機代数の教科書 [9] [12] や computational group theory の教科書 [4], オンラインのテキスト [2] などで紹介されているアルゴリズムの実装を目的に書いている JavaScript ライブラリのことである。「素朴な JavaScript」とは、emscripten などを利用し、C や C++ で書かれたコードを JavaScript に変換したものではなく、直接テキストファイルに人間が書いた JavaScript という意味である。著者の現時点の到達点は、Gebauer-Möller による groebner 基底の導出、Pell 方程式、ユークリッド環 (整数  $\mathbb{Z}$  や体  $K$  上の1変数多項式環  $K[x]$ ) 上の行列の単因子計算、Berlekamp algorithm による  $\mathbb{F}_p[x]$  における因数分解、Hensel 構成による  $\mathbb{Z}[x]$  における因数分解、Schreier-Sims のアルゴリズムによる置換群の元の生成、Minkwitz による任意の元を、与えられた生成元の積へ分解するアルゴリズムなどである。昨年、教科書 [9] を元に Gebauer-Möller を実装し、試しに、教科書 [7] p82 2.40 [例] を辞書式順序で解かせてみたところ、「性能の低いソフトでは、相当待っても終わらない」とのことであるが、Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz 搭載の ubuntu 17.10 ノート PC で、約 14 秒ほどで被約 groebner 基底を導出することができた。以上により、JavaScript は、コーディングの洗練によって、大学レベルの数学、特に可換環論などの代数学、整数論、置換群、組合せ論などの教科書レベルの実例計算アルゴリズムを実装する上で、十分な性能を有する言語であると著者は考えている。

Jordan 標準形の実例計算を実装する上で、実際に必要となった主要な計算代数アルゴリズムを列挙すると、

- ユークリッド環上の単因子構成アルゴリズム
- Berlekamp アルゴリズムからの Hensel 構成による因数分解アルゴリズム
- 拡張ユークリッド互除法による中国剰余構成アルゴリズム

となる。

## 3 Jordan 標準形

### 3.1 単因子論の基本

最初に、単因子論の有名な基本的事実を [10] に従ってまとめておく。非数学科出身者にもわかり易く解説するという動機に基づく研究なので、初歩的な内容であっても丁寧に説明し、個々に実例を対比させていく必要がある。可換環  $R$  上  $m$  行  $n$  列行列全体の

集合を  $\text{Mat}_{m,n}(R)$  と表す.  $m$  次正方行列全体の集合は  $\text{Mat}_m(R)$  と表す. また,  $R$  上可逆な  $n$  次行列全体の集合 (一般線形群) を  $\text{GL}_n(R)$  と表す.  $x$  を不定元とする 1 変数多項式環  $R[x]$  とする.

**定理 3.1.** 単項イデアル整域  $R$  上の  $m$  行  $n$  列行列  $A \in \text{Mat}_{m,n}(R)$  について, ある  $P \in \text{GL}_m(R)$  とある  $Q \in \text{GL}_n(R)$  が存在し,

$$PAQ = \left[ \begin{array}{ccc|c} e_1 & & & \\ & \ddots & & \\ & & e_r & \\ \hline & & & 0 \end{array} \right]$$

ここで,  $e_1|e_2|\cdots|e_r$ . この  $e_1, e_2, \dots, e_r$  を  $A$  の単因子という.

**系 3.1.** 体  $K$  上 1 変数多項式環  $K$  上の  $m$  次正方行列  $A \in \text{Mat}_m(K)$  について, ある  $P(x), Q(x) \in \text{GL}_m(K[x])$  が存在し,

$$P(x)(xI_m - A)Q(x) = \begin{bmatrix} e_1(x) & & \\ & \ddots & \\ & & e_n(x) \end{bmatrix}$$

ここで, 各  $e_i(x)$  はモニックであり,  $e_1(x)|e_2(x)|\cdots|e_n(x)$ . 特に,  $e_1(x)e_2(x)\cdots e_n(x)$  は  $A$  の特性多項式であり,  $e_n(x)$  は  $A$  の最小多項式である.

**定理 3.2.** 体  $K$  上の  $m$  次正方行列  $A, B \in \text{Mat}_m(K)$  について次の条件は同値である.

- (1) ある  $P(x), Q(x) \in \text{GL}_m(K[x])$  が存在し,  $(xI_m - A)P(x) = Q(x)(xI_m - B)$
- (2)  $xI_m - A$  と  $xI_m - B$  の単因子は一致する.
- (3) ある  $P \in \text{GL}_m(K)$  が存在し,  $B = P^{-1}AP$

条件 (1) が成り立つとき,  $P(x) = P_1(x)(xI_m - B) + P$  を満たす  $P_1(x) \in \text{Mat}_m(K[x])$  と  $P \in \text{GL}_m(K)$  が存在し,

$$B = P^{-1}AP$$

## 3.2 Jordan 標準形の実装

Jordan 標準形の実装に必要な数学を, F.M.Goodman が公開しているテキスト [6] p.405 に倣って述べる. Jordan 標準形の導出のみではなく,  $P^{-1}AP$  を Jordan 標準形にするような変換行列  $P$  を具体的に導出するための実装までが目標である. 理論の見通しを良くするため, 体  $K$  上  $n$  次元ベクトル空間  $V$  を一つとり,  $V$  の  $K$  ベクトル空間としての基底も一つとり, それを  $\{e_1, \dots, e_n\}$  とする.  $V$  上の線形変換全体の集合を

$\text{End}_K(V)$  と表す.  $T \in \text{End}_K(V)$  に対し,  $T$  を表現する行列を  $A \in \text{Mat}_n(K)$  とする. 即ち

$$(T(e_1), \dots, T(e_n)) = (e_1, \dots, e_n)A$$

$K[x]$  の  $V$  への作用を  $xv = T(v)$  ( $\forall v \in V$ ) と定義することによって,  $V$  は  $K[x]$ -加群となる. 階数が  $n$  である  $K[x]$  上の自由加群  $F$  をとり, その基底  $\{f_1, \dots, f_n\}$  を一つとるとき,  $K[x]$ -加群としての準同型写像  $\Phi: F \rightarrow V$  で  $\Phi(f_i) = e_i$  ( $\forall i = 1, 2, \dots, n$ ) なるものが定義できる.  $\tilde{T} \in \text{End}_{K[x]}(F)$  を

$$(\tilde{T}(f_1), \dots, \tilde{T}(f_n)) = (f_1, \dots, f_n)A$$

となるものと定義すると, 明らかに

$$\begin{array}{ccc} F & \xrightarrow{\Phi} & V \\ \tilde{T} \downarrow & & \downarrow T \\ F & \xrightarrow{\Phi} & V \end{array}$$

なる  $K[x]$ -加群としての可換図式が得られる. このとき

**補題 3.1.**

$$\text{Im}(x\text{Id}_F - \tilde{T}) = \text{Ker}\Phi$$

*Proof.* 実際, 任意の  $f \in F$  に対し  $\Phi(xf) = T\Phi(f) = \Phi(\tilde{T}f)$  より,  $\Phi((x\text{Id}_F - \tilde{T})(f)) = 0$  となるので,  $\text{Im}(x\text{Id}_F - \tilde{T}) \subseteq \text{Ker}\Phi$ . また,  $F_0 := \bigoplus_{i=1}^n Kf_i$  とおくと,

$$F = \text{Im}(x\text{Id}_F - \tilde{T}) + F_0$$

が示される. これは簡単な計算より, 任意の  $h(x) \in K[x]$  に対し,  $h(x) - h(y) = (x - y)g(x, y)$  ( $\exists g(x, y) \in K[x, y]$ ) となることから, 任意の  $i = 1, 2, \dots, n$  に対し,

$$h(x)f_i = (h(x)\text{Id}_F - h(\tilde{T}))(f_i) + h(\tilde{T})(f_i) = (x\text{Id}_F - \tilde{T})g(x\text{Id}_F, \tilde{T})(f_i) + h(\tilde{T})(f_i)$$

であることから分かる. また,

$$\text{Ker}\Phi \cap F_0 = 0$$

であることも,  $f \in \text{Ker}\Phi \cap F_0$  とするとき,  $f = \sum_{i=1}^n a_i f_i$  ( $\exists a_1, \dots, a_n \in K$ ) となり,  $0 = \Phi(f) = \sum_{i=1}^n a_i e_i$  より,  $a_1 = \dots = a_n = 0$  となることから分かる. 以上により

$$F = F_0 \oplus \text{Ker}\Phi, \quad \text{Ker}\Phi = \text{Im}(x\text{Id}_F - \tilde{T})$$

□

ここで,  $w_j := (x\text{Id}_F - \tilde{T})(f_j) = xf_j - \sum_{i=1}^n a_{ij}f_i$  ( $j = 1, 2, \dots, n$ ) とおくと,

$$(w_1, \dots, w_n) = (f_1, \dots, f_n)(xI_n - A)$$

となり, 特に,  $\{w_1, \dots, w_n\}$  が  $\text{Ker}\Phi$  の  $K[x]$ -加群としての基底となることが分かる. 系 3.1 により,

$$P(x)(xI_n - A)Q(x) = D(x), \quad D(x) = \begin{bmatrix} e_1(x) & & \\ & \ddots & \\ & & e_n(x) \end{bmatrix}$$

となるような  $P(x), Q(x) \in \text{GL}_n(K)$  とモニツク多項式  $e_1(x), \dots, e_n(x) \in K[x]$  が存在し,  $e_1(x)|e_2(x)|\dots|e_n(x)$  となる.

$$e_1(x) = \dots = e_{n-s}(x) = 1 \neq e_{n-s+1}$$

となるような  $s$  が存在するので,  $a_i(x) := e_{n-s+i}$  ( $i = 1, \dots, s$ ) とおき,

$$(y_1, \dots, y_{n-s}, z_1, \dots, z_s) := (f_1, \dots, f_n)P^{-1}(x)$$

とおくと, これは  $F$  の  $K[x]$ -加群としての基底である. 特に,

$$\{\Phi(y_1), \dots, \Phi(y_{n-s}), \Phi(z_1), \dots, \Phi(z_s)\}$$

は  $V$  の  $K[x]$ -加群としての生成系となる.

$$\begin{aligned} (w_1, \dots, w_n)Q(x) &= (f_1, \dots, f_n)(xI_n - A)Q(x) \\ &= (f_1, \dots, f_n)P(x)^{-1} P(x)(xI_n - A)Q(x) \\ &= (y_1, \dots, y_{n-s}, z_1, \dots, z_s)D(x) \\ &= (y_1, \dots, y_{n-s}, a_1(x)z_1, \dots, a_s(x)z_s) \end{aligned}$$

となる. これは  $\text{Ker}\Phi$  の  $K[x]$ -加群としての基底であるので,

$$\Phi(y_1) = \dots = \Phi(y_{n-s}) = 0$$

であり, 更に  $v_i := \Phi(z_i)$  ( $i = 1, \dots, s$ ) とおくと,  $\{v_1, \dots, v_s\}$  は  $V$  の  $K[x]$ -加群としての生成系であり,

$$a_1(x)v_1 = \dots = a_s(x)v_s = 0$$

となることが分かる.  $n_i := \deg(a_i(x))$  とするとき,  $K[x]v_i = \sum_{j=0}^{n_i-1} Kx^jv_i$  であり,  $\sum_{i=1}^s n_i = n$  だから, 次元の比較により

$$K[x]v_i = \bigoplus_{j=0}^{n_i-1} Kx^jv_i$$

$$V = \bigoplus_{i=1}^s K[x]v_i$$

となる.

$$\varphi_i : K[x]/K[x]a_i(x) \rightarrow K[x]v_i$$

を  $\varphi_i(1) = v_i$  となる  $K[x]$ -加群の準同型とすると, これは同型写像であることが分かる.  $(v_i, xv_i, \dots, x^{n_i-1}v_i)$  で  $x$  の作用を表現すれば, 有名な有理標準形と呼ばれる行列になる. 例えば,  $a_i(x) = \sum_{j=0}^{n_i} c_j^{(i)} x^j$  はモニック (i.e.  $c_{n_i}^{(i)} = 1$ ) だから,

$$x(v_i, xv_i, \dots, x^{n_i-1}v_i) = (v_i, xv_i, \dots, x^{n_i-1}v_i)C(a_i(x))$$

となる. 但し,

$$C(a_i(x)) := \begin{bmatrix} & & -c_0^{(i)} \\ & & -c_1^{(i)} \\ & 1 & -c_2^{(i)} \\ & & \ddots \\ & & & \vdots \\ & & & 1 & -c_{n_i-1}^{(i)} \end{bmatrix} \in \text{Mat}_{n_i}(K)$$

とする.  $v_i$  を列ベクトルとみなして,

$$P := (x^j v_i \mid i \in \{1, \dots, s\}, j \in \{0, 1, \dots, n_i - 1\})$$

とおくと,

$$P^{-1}AP = C(a_1(x)) \oplus \dots \oplus C(a_s(x))$$

と行列の直和分解となる. 目標としているのは, Jordan 標準形と, その変換行列の導出過程のプログラミングによる可視化であるので, 以降は,

$A$  の特性多項式  $\det(xI_n - A)$  は,  $K[x]$  において 1 次式の積に因数分解される

と仮定する.

$a_1(x)|a_2(x)|\dots|a_s(x)$  なので,  $a_s(x)$  の解を  $\alpha_1, \dots, \alpha_t$  とすると,

$$a_i(x) = \prod_{j=1}^t (x - \alpha_j)^{\lambda_{ij}}$$

であり, 任意の  $j = 1, \dots, t$  に対して

$$\lambda_{1j} \leq \lambda_{2j} \leq \dots \leq \lambda_{sj}$$

更に,  $\sum_{ij} \lambda_{ij} = n$  となる. Jordan 標準形は  $V$  の各直和因子  $K[x]v_i$  に対し, 中国剰余定理を適用して構成した  $K$ -ベクトル空間としての基底による行列表現となって現れる. 具体的には, 拡張ユークリッド互除法によって, 各  $j = 1, \dots, t$  と  $k \neq j$  に対し,

$$\sigma_{jk}(x)(x - \alpha_j)^{\lambda_{ij}} + \tau_{jk}(x)(x - \alpha_k)^{\lambda_{ik}} = 1$$

となるような  $\sigma_{jk}(x), \tau_{jk}(x) \in K[x]$  が構成できるので,

$$\varepsilon_j^{(i)}(x) := \prod_{k \neq j} (1 - \sigma_{jk}(x)(x - \alpha_j)^{\lambda_{ij}}) = \prod_{k \neq j} \tau_{jk}(x)(x - \alpha_k)^{\lambda_{ik}}$$

とおくと,

$$\varepsilon_j^{(i)} \equiv \begin{cases} 1 & \text{mod } (x - \alpha_j)^{\lambda_{ij}} \\ 0 & \text{mod } (x - \alpha_k)^{\lambda_{ik}} \quad (\text{if } k \neq j) \end{cases}$$

となる. 故に,

$$K[x]v_i = \bigoplus_{j=1}^t K[x]\varepsilon_j^{(i)}v_i = \bigoplus_{j=1}^t \bigoplus_{l=0}^{\lambda_{ij}-1} K(x - \alpha_j)^l \varepsilon_j^{(i)}v_i$$

であり, この  $K[x]v_i$  の  $K$ -ベクトル空間としての基底で  $x$  を表現すれば,

$$\bigoplus_{j=1}^t J_{\lambda_{ij}(\alpha_j)}$$

ただし,

$$J_\lambda(\alpha) := \begin{bmatrix} \alpha & & & & \\ 1 & \alpha & & & \\ & 1 & \alpha & & \\ & & & \ddots & \\ & & & & \alpha \\ & & & & 1 & \alpha \end{bmatrix} \in \text{Mat}_\lambda(K)$$

と定義し, これを  $\alpha$  に対する  $\lambda$  次の Jordan ブロックという.

著者は `ca.js` の計算を試すことのできる web サイト [3] において, 「13 matrices over the ring of polynomials in one variable over the rational field」の「03 Jordan canonical form」に, ここで述べられた議論の実例の計算を, ランダムに生成し, 計算結果を導出する過程を「可視化」を実装した. 体  $K$  は有理数体  $\mathbb{Q}$  に限っている. 以下に, その「可視化」の流れを述べる. 始めに解答を作成し, 例えば

$$J = \begin{bmatrix} -\frac{3}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{3}{2} & 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{3}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{5} \end{bmatrix} \in \text{Mat}_6(\mathbb{Q})$$

とする. 可逆行列  $P$  をランダムに生成する. 実際に,

$$P = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 3 & 1 & 1 \\ 0 & 2 & 3 & -2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \in \text{GL}_6(\mathbb{Q})$$



が生成される.

$$A := PJP^{-1} = \begin{bmatrix} -\frac{3}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{17}{10} & -\frac{49}{10} & -\frac{153}{10} & -\frac{51}{10} & \frac{153}{10} & 0 \\ -\frac{7}{10} & \frac{7}{5} & \frac{17}{2} & \frac{31}{10} & -9 & \frac{7}{10} \\ \frac{2}{5} & -\frac{4}{5} & -\frac{177}{10} & -\frac{57}{10} & \frac{114}{5} & -\frac{51}{10} \\ 0 & 0 & -1 & 0 & \frac{11}{5} & -1 \\ \frac{7}{10} & -\frac{7}{5} & -\frac{103}{10} & -\frac{31}{10} & 13 & -\frac{5}{2} \end{bmatrix} \in \text{Mat}_6(\mathbb{Q})$$

とおき, この  $A$  の Jordan 標準形を求める過程を生成する仕掛けとなっている.  $xI_6 - A$  の Smith 標準形を計算させると,

$$P(x)(xI - A)Q(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & x^2 + \frac{13}{10}x - \frac{3}{10} & 0 \\ 0 & 0 & 0 & 0 & 0 & x^4 + \frac{13}{5}x^3 + \frac{109}{100}x^2 - \frac{39}{50}x + \frac{9}{100} \end{bmatrix}$$

となるような  $P(x), Q(x) \in \text{GL}_6(\mathbb{Q}[x])$  が計算される. この  $P(x)^{-1}$  の列ベクトルを  $K[x]$ -自由加群  $F$  としての基底だから,  $\Phi: F \rightarrow V$  の像を取ると,  $V$  の  $K[x]$ -加群としての生成系となるのであった.  $P(x)^{-1}$  は複雑でありスペースの関係上ここに載せることはできないが, 著者のサイト [3] では見ることができる.

$$\Phi(P(x)^{-1}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \\ -2 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} -31 \\ -10 \\ -17 \\ 54 \\ 0 \\ 0 \end{bmatrix}$$

という結果が返ってくる. 0 でない最後の 2 列が  $V$  の  $K[x]$ -加群としての生成系となる. これらと中国剰余を使って計算させると,  $V$  の  $K$ -加群としての基底が得られるが, これも, スペースの関係上ここに載せることはできないが, 著者のサイト [3] では見ることができる. 以下の  $P_0$  の列ベクトルが  $V$  の  $K$ -ベクトルとしての基底である.

$$P_0 = \begin{bmatrix} 2 & 0 & -31 & 0 & 0 & 0 \\ 1 & -1 & 10 & 0 & -20 & 0 \\ 0 & 2 & -20 & -17 & 3 & 17 \\ 0 & -2 & 43 & 51 & 11 & 0 \\ 0 & 1 & 0 & 0 & 0 & 17 \\ 0 & 0 & 20 & 17 & -20 & 17 \end{bmatrix}, P_0^{-1} = \begin{bmatrix} -\frac{7}{17} & \frac{31}{17} & \frac{93}{17} & \frac{31}{17} & -\frac{93}{17} & 0 \\ -1 & 2 & \frac{173}{17} & 3 & -\frac{193}{17} & \frac{20}{17} \\ -\frac{1}{17} & \frac{2}{17} & \frac{6}{17} & \frac{2}{17} & -\frac{6}{17} & 0 \\ \frac{3}{289} & -\frac{6}{289} & \frac{33}{289} & \frac{11}{289} & -\frac{50}{289} & \frac{1}{17} \\ 0 & 0 & -\frac{1}{17} & 0 & \frac{2}{17} & -\frac{1}{17} \\ \frac{1}{17} & -\frac{2}{17} & -\frac{173}{289} & -\frac{3}{17} & \frac{210}{289} & -\frac{20}{289} \end{bmatrix}$$

となって,

$$P_0^{-1}AP_0 = \begin{bmatrix} -\frac{3}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{3}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & -\frac{3}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{5} \end{bmatrix}$$

## 4 まとめ

因数分解アルゴリズムは 30 次くらいでは一瞬なので, あまり気にする必要はないが, 行列の次数が大きくなれば, 単因子の計算は合理的な時間で計算することはできない. 例えば, 計算時間が 8 次になると 30 秒近くなり, 9 次になると 1 分を超えることがある. 今後の課題は, 単因子の導出速度の向上である.

## 参考文献

- [1] Peter Olson : BigInteger.js, <https://www.npmjs.com/package/big-integer>
- [2] 野呂正行 : 計算機代数入門, <http://www.math.kobe-u.ac.jp/Asir/ca.pdf>
- [3] ca.js, [https://noble-garden-math.jp/math/web\\_note/computer\\_algebra.php](https://noble-garden-math.jp/math/web_note/computer_algebra.php)
- [4] Gregory Butler : Fundamental Algorithms for Permutation Groups (Lecture Notes in Computer Science). Springer. 1991
- [5] 牛腸徹 : Jordan 標準形に対する単因子論を用いたアプローチについて, <https://lecture.ecc.u-tokyo.ac.jp/~nkiyono/gocho/en14apx.pdf>
- [6] F.M.Goodman, ALGEBRA ABSTRACT AND CONCRETE EDITION 2.6 : <http://homepage.math.uiowa.edu/~goodman/algebrabook.dir/book.2.6.pdf>
- [7] 丸山正樹 : グレブナー基底とその応用 (共立叢書・現代数学の潮流). 共立出版. 2002
- [8] Math & Coding : <https://math-coding.connpass.com>
- [9] 野呂正之・横山和弘 : グレブナー基底の計算 基礎篇 計算代数入門. (財) 東京大学出版会. 2002
- [10] 杉浦光夫, 横沼健雄 : 「ジョルダン標準形・テンソル代数」, 岩波書店, 1990.
- [11] 川久保勝夫 : 「線形代数学」, 日本評論社, 2010.
- [12] 和田秀男 : 数の世界—整数論への道 (科学ライブラリー). 岩波書店. 1981